

University of Regina
Faculty of Engineering and Applied Science
Software Systems Engineering

ENSE 374 - Software Engineering Management - Laboratory

Instructor:

Mr. Adam Tilson, M.A.Sc., Engineer-in-Training

Adam.Tilson@uregina.ca

Zoom office hours by appointment, please e-mail to inquire.

Lab Scheduling:

Term: Fall 2021

Section 091. Tuesday. 2:30 pm – 5:15 pm. Zoom.

Section 092. Wednesday. 11:30 am – 2:15 pm. Zoom.

Labs will not run every week – please check the schedule.

Lab Overview:

In this lab you will apply the principles learned in class to incrementally develop a client / server architecture using the Node.js stack and the MVC Architectural Pattern. Components include:

- Implement View with HTML, CSS, Bootstrap, jQuery, EJS
- Implement Controller with Node.js and Express.js
- Implement Model with MongoDB, Mongoose and Passport

Schedule and Grading:

	Assigned	Due	Grading
1. Tools: VS Code, Git and GitHub	Sept 7/8	Sept 14/15 +	1%
2. View 1: HTML	Sept 14/15	Sept 21/22 +	2%
3. View 2: CSS & Bootstrap	Sept 21/22	Sept 28/29	4%
4. View 3: JavaScript and DOM	Sept 28/29	Oct 5/6 +	2%
5. View 4: jQuery	Oct 5/6	Oct 12/13	4%
6. Controller 1: Node.js	Oct 12/13	Oct 19/20 +	2%
7. Controller 2: Express & EJS	Oct 19/20	Oct 26/27	4%
8. Model 1: MySQL and MongoDB	Oct 26/27	Nov 2/3 +	2%
9. Model 2: Mongoose and Passport	Nov 2/3	Nov 16/17	4%
Total Lab Grade:			25% *

(+) labs will be graded by interview. Others will be submitted through URCourses.

* Grading subject to adjustment at the instructor's discretion (typically +/- 1% per lab)

Lab Topics:

1: Tools

Installing and using Visual Studio Code, and exploring some of the best features. Installing Git, and an overview of version control. Understanding GitHub. Creating good GitHub projects, pages and using GitHub as a code portfolio.

2. View 1: HTML

A primer on HTML. We will cover web page layout composition and common tags.

3. View 2: CSS and Bootstrap

A primer on CSS, and an introduction to the Bootstrap library to make CSS development faster and more consistent. This lab will have a hand-in assignment based on implementing and styling the skeleton of the View component of our lab project.

4. View 3: JavaScript

A primer on the JavaScript programming language, and how it can be used to interact with an HTML page through the document object model (DOM).

5. View 4: jQuery

A primer on the jQuery library to speed up JavaScript development and DOM manipulation. This lab will have a hand-in assignment based on creating an interactive front-end using JavaScript, though these will mostly be replaced later.

6. Controller 1: Node.js

This lab will introduce the MVC architecture, and introduce Node.js for creating a Client / Server architecture on the local machine.

7. Controller 2: Express.js and EJS

This lab will introduce Express.js for simplifying Node.js server operations, and EJS for creating View templates. This lab will have a hand-in assignment based on serving our web page using Node.js and dynamically generating content.

8. Model 1: MySQL and MongoDB

A primer on databases by first looking at MySQL, which is more structured and easier to learn, and then examining MongoDB which is commonly used in the Node.js Stack.

9. Model 2: Mongoose and Passport

This lab completes the Model using the MongoDB extensions Mongoose and Passport, adding persistent storage to the application, and creating and authenticating users using Cookies and Sessions. This lab will have the final hand-in assignment which completes the lab project.

Lab Policies:

Submission Policy:

Several labs will be graded by demonstration and interview over Zoom. The student will meet with the instructor in a breakout room, show their solution with screen sharing, and answer a few questions. These demos may be performed the week the lab is assigned, or the following week with no penalty.

The remaining assignments will be submitted through URCourses. The hand-in assignments are iterative in nature, so it is required one is completed before the next can be started.

Late Submissions:

Once per semester you are granted a grace week for submissions, and may submit up to a week late, up to the last day of classes, with no penalty. You do not need to provide any rationale, but do let me know you are using your grace week for this assignment.

Group Work:

All lab work in this course must be done individually. Students may collaborate at a very high level, e.g. sharing general strategies to problems, but no code may be shared.

All code submitted, excluding provided code, must be your own.

Academic Misconduct:

Exception: Brief snippets of others' code from tutorials are permitted, e.g. from StackOverflow, but only if:

1. It does not solve a core part of the assigned problem, and
2. It is cited with a URL and brief explanation in a comment.

Don't submit code you don't understand.

If in doubt – learn it and write it yourself!

Cheating is a serious offense. Students caught cheating will be subject to the disciplinary proceedings in the University Calendar. Consequences of academic misconduct may include receiving a 49% in the course, or receiving a grade of XF (academic misconduct) for the course, recorded permanently on the student's transcript. (See Section 5.14 in the University Calendar.) Further penalties may be imposed by APEGS.

Accessibility:

If you require special accommodations due to a disability, medical situation or other extenuating circumstances, discuss this with the instructor in the first week of labs. Register with Centre for Student Accessibility at accessibility@uregina.ca (See Section 8.2.15 of the University Calendar)

Zoom Information:

To be provided on URCourses prior to the lab.